

Eusko Jaurlaritzaren Informatika Elkartea
Sociedad Informática del Gobierno Vasco

TortoiseSVN:

Manual de usuario

Fecha:

Referencia:

EJIE S.A.

Mediterráneo, 3

Tel. 945 01 73 00*

Fax. 945 01 73 01

01010 Vitoria-Gasteiz

Posta-kutxatila / Apartado: 809

01080 Vitoria-Gasteiz

www.ejie.es

Control de documentación

Título de documento: TortoiseSVN

Histórico de versiones

Código:

Versión: 1.0

Fecha:

Resumen de cambios:

Cambios producidos desde la última versión

Primera versión.

Control de difusión

Responsable:

Aprobado por:

Firma:	Fecha:

Distribución:

Referencias de archivo

Autor: Consultoría de áreas de conocimiento

Nombre archivo: TortoiseSVN. Manual de usuario.doc

Localización:

Contenido

1 Introducción

El presente documento describe cuáles son las tareas básicas que se pueden ejecutar en la explotación de la herramienta TortoiseSVN.

2 Conceptos básicos

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones Subversion. Está desarrollado bajo la Licencia Pública General GNU (GPL) y se puede acceder al código fuente de este programa en este enlace http://tortoisesvn.tigris.org/svn/tortoisesvn/.

TortoiseSVN maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio.

Entre las características de TortoiseSVN cabe destacar:

Integración con el shell de Windows
TortoiseSVN se integra perfectamente en el shell de Windows (por ejemplo, el explorador). Esto significa que puede seguir trabajando con las herramientas que ya conoce. ¡Y que no tiene que cambiar a una aplicación diferente cada vez que necesite las funciones del control de versiones!

Los menús contextuales de TortoiseSVN también funcionan en otros administradores de archivos, y en el diálogo Archivo/Abrir que es común a la mayoría de aplicaciones estándar de Windows. Sin embargo, debe tenerse en cuenta que TortoiseSVN está desarrollado con la mirada puesta en hacerle extensión del Explorador de Windows. Por este motivo, puede que en otras aplicaciones la integración no sea tan completa y que, por ejemplo, los iconos sobreimpresionados en las carpetas no se muestren.

Iconos sobreimpresionados
El estado de cada carpeta y fichero versionado se indica por pequeños iconos sobreimpresionados. De esta forma, puede ver fácilmente el estado en el que se encuentra su copia de trabajo.



Fácil acceso a los comandos de Subversion
Todos los comandos de Subversion están disponibles desde el menú contextual del explorador. TortoiseSVN añade su propio submenú allí.

TortoiseSVN proporciona una serie de **herramientas internas**:

- **TortoiseMerge:** Permite ver las diferencias entre ficheros de texto, fusionar esos cambios e incluso revisar y aplicar ficheros de diff unificados, a menudo llamados *parches*.
- **TortoiseBlame:** Hace más fácil la lectura de los ficheros de autoría.

- **SubWCRev :** Es un programa de consola para Windows que puede utilizarse para leer el estado de una copia de trabajo local y opcionalmente realizar sustituciones de palabras clave en un fichero plantilla.

3 ¿Por qué un sistema de trabajo en grupo?

En muchas ocasiones, un programador habrá modificado un programa en el cual han aparecido inconsistencias y ha tenido que volver atrás, teniendo que recordar cómo estaba antes, dónde, etc. Con un sistema de control de versiones restaurar a un estado anterior es tan fácil que cuando se usa por primera vez será algo indispensable en proyectos futuros.

Usando las herramientas de cliente de sistemas como Subversion, los desarrolladores pueden realizar cambios al contenido del repositorio. El repositorio será algo así como una biblioteca central donde se guarda todo el código de un proyecto. Se puede cambiar cada archivo, creando una historia completa de la evolución del proyecto en desarrollo; los desarrolladores pueden recuperar versiones más antiguas de un archivo, ver un registro de cambios, y realizar otras tareas útiles según lo necesitado. Subversion está dirigido al trabajo en grupo, aunque también es muy útil para un programador individual por la posibilidad de recuperar viejas copias y ver los cambios hechos.

Subversion es un sistema centralizado para compartir información. En su interior, es un repositorio, es decir, un almacén de información. El repositorio almancena información en forma de árbol. Utiliza una jerarquía típica de ficheros y directorios. Cuando un desarrollador añade archivos con código, ésta pasa a ser disponible para otros. De hecho, se pueden conectar varios clientes a ese repositorio y entonces leer y escribir archivos en el mismo.

En principio, parece un sistema muy similar a un típico servidor de ficheros. Pero en realidad, más que un servidor en sí, lo que le hace interesante a Subversion es que es capaz de “recordar” cualquier cambio realizado sobre un archivo de código:

- Cualquier cambio a un archivo.
- Cambios en un directorio.
- Archivos que se añaden, se borran, se modifican, etc.
- Cuándo se realizan los cambios.
- Quién los realiza.

Cuando un cliente lee datos de un repositorio, normalmente ve sólo la última versión del sistema de árbol. Pero el cliente tendrá la posibilidad de buscar estados previos de dicho sistema. Por ejemplo, un cliente puede hacer preguntas del tipo: “¿Qué contenía este directorio hace 4 días?” o “¿Quién ha sido la última persona que ha modificado este fichero y qué cambios hizo?”

Éstas son las típicas preguntas que se pueden resolver con un sistema de repositorio de archivos como Subversion, ya que almacenan toda la información tanto de versiones, cambios por versión y responsable de dichos cambios.

Debido a todo lo anterior, permite además la concurrencia en el trabajo, es decir, que más de una persona pueda trabajar con el mismo proyecto y, lo que es más importante, incluso con el mismo archivo.

4 Funciones elementales

Antes de empezar a utilizar TortoiseSVN se deben cumplir las siguientes premisas:

- Tener ya instalado TortoiseSVN.

- Estar familiarizado con los sistemas de control de versiones.

- Conocer las bases de Subversion.
- Tener preparado un servidor y/o tener acceso a un repositorio de Subversion.

También es interesante conocer el tratamiento que hace Subversion a la rama principal (trunk), ramas (branches) y etiquetas (tags o releases):

- Una de las características de los sistemas de control de versiones es la posibilidad de aislar cambios en una línea separada de desarrollo. Esto se conoce como una rama. Las ramas se utilizan a menudo para probar nuevas características sin molestar la línea principal del desarrollo con errores de compilación y errores. Tan pronto como la nueva característica es lo suficiente estable, la rama de desarrollo se fusiona de nuevo en la rama principal (trunk, troncal).
- Otra característica de los sistemas de control de versiones es la posibilidad de marcar revisiones particulares (por ejemplo, una versión lanzada a producción), para que pueda en cualquier momento recrear un cierto entorno o compilación. Este proceso se conoce como etiquetar (tags o releases).

Importante respecto a CVS: Dejan de existir los conceptos de módulo, ramas, y etiquetas, como entidades separadas:

- Ramas y etiquetas (branches y tags): Internamente no existen tales conceptos, si quieres una rama la creas a partir de un enlace a un número de revisión del código (se comparte el historial, no es necesario copiar el proyecto entero). Una etiqueta es una rama a la que no se le añaden más cambios.
- Módulos: En Subversion solo existen directorios, no hay concepto de módulo.

Antes de que se importen datos al repositorio, primero se deberá pensar cómo se quiere organizar los datos. Si se utiliza uno de los patrones recomendados será más fácil. Hay algunas formas estándar y recomendadas de organizar un repositorio. La mayoría de la gente crea un directorio trunk (tronco) para alojar la "línea principal" del desarrollo, un directorio branches (ramas) para que contenga las copias/ramas, y un directorio tags (etiquetas) para contener las copias/etiquetas.

Si un repositorio sólo aloja un proyecto, entonces la gente a menudo crea estos directorios en el primer nivel:

```
/trunk
/branches
/tags
```

Si un repositorio contiene múltiples proyectos, la gente a menudo indexa por ramas:

```
/trunk/paint
/trunk/calc
/branches/paint
/branches/calc
/tags/paint
/tags/calc
```

o por proyecto:

```
/paint/trunk
/paint/branches
```

```
/paint/tags
/calc/trunk
/calc/branches
/calc/tags
```

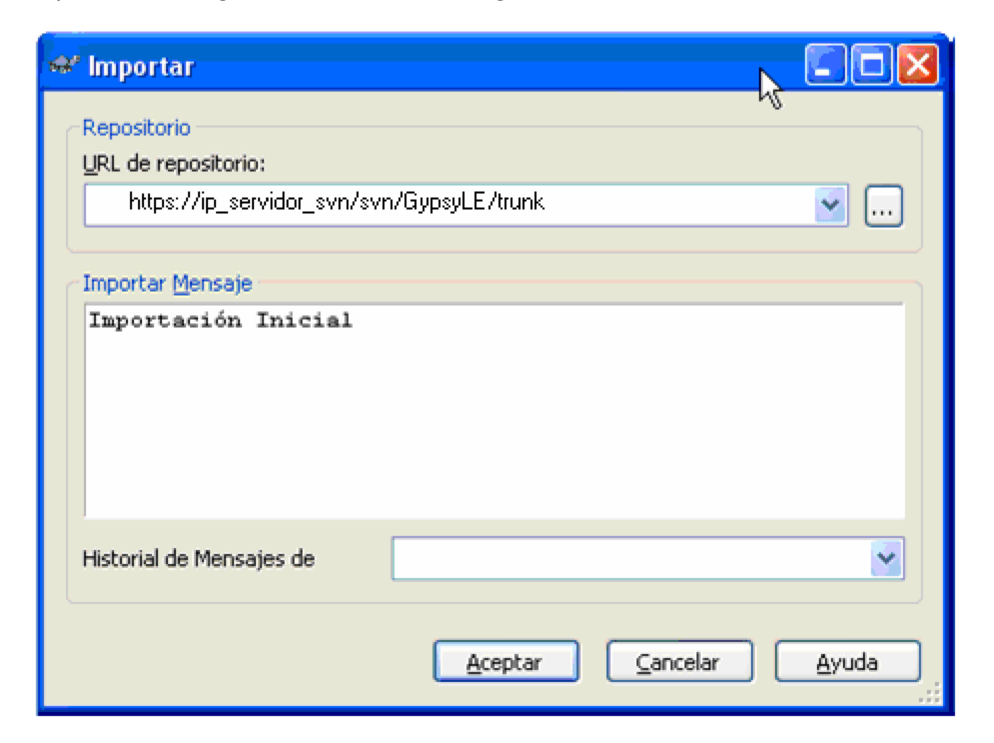
4.1 Importar datos en un repositorio

1. Seleccionar la carpeta superior de la estructura de directorios del proyecto en el explorador de Windows.

2. Hacer "clic" con el botón derecho del ratón para abrir el menú contextual.

3. Seleccionar el comando *TortoiseSVN > Importar...*

4. Aparecerá el siguiente cuadro de diálogo:



5. Introducir la URL del repositorio en el que quiere importar su proyecto. El mensaje de importación se utiliza como un mensaje de registro.

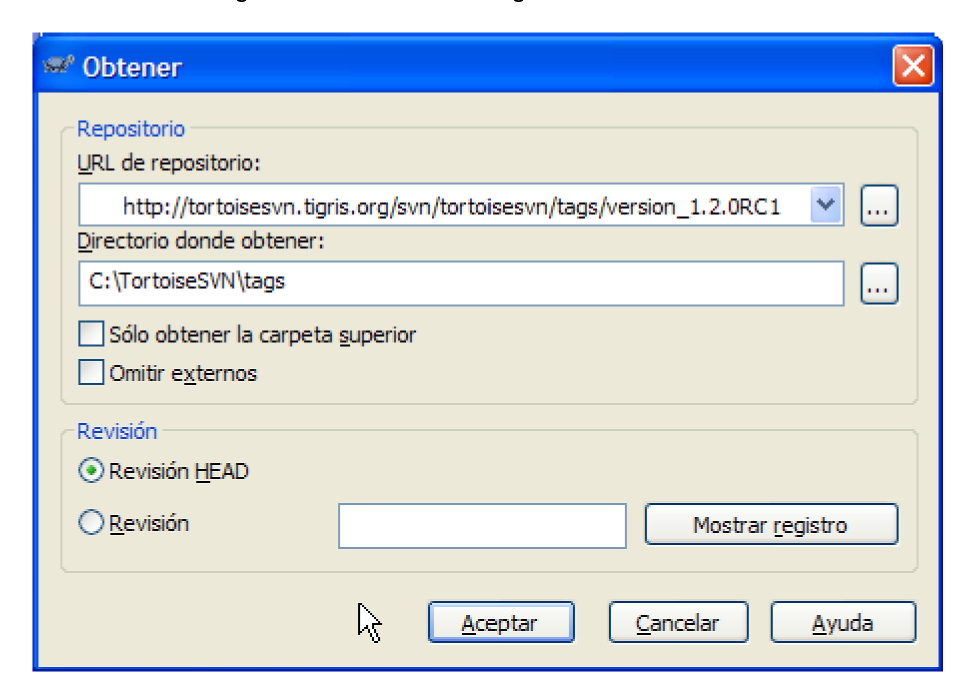
6. Tan pronto como se presione "Aceptar", TortoiseSVN importa el árbol completo de directorios incluyendo todos los ficheros en el repositorio. El nombre de la carpeta que se está importando no aparece en el repositorio, sólo los contenidos de los ficheros. El proyecto ahora está almacenado en el repositorio bajo el control de versiones.

NOTA: Hay que tener en cuenta que la carpeta que se ha importado no está bajo el control de versiones. Para obtener una copia de trabajo bajo el control de versiones se necesita obtener la versión que se acaba de importar.

4.2 Obtener una copia de trabajo

Para conseguir una copia de trabajo se necesita obtener una de un repositorio. Para ello:

1. Seleccionar un directorio en el explorador de Windows donde se quiera poner la copia de trabajo.
2. Hacer "clic" con el botón derecho del ratón para mostrar el menú contextual.
3. Seleccionar el comando *TortoiseSVN > Obtener...*
4. Se mostrará el siguiente cuadro de diálogo:



5. Antes de pulsar “Aceptar” habrá que tener en cuenta que:
   - Si se introduce un nombre de carpeta que aún no exista, se creará un directorio con ese nombre.
   - Si se desea obtener sólo la carpeta superior y omitir todas las subcarpetas, habrá que hacer uso de la casilla “Sólo obtener la carpeta superior”.
   - Si el proyecto contiene referencias a proyectos externos que no se desea que se obtengan al mismo tiempo, habrá que hacer uso de la casilla “Omitir externos”.

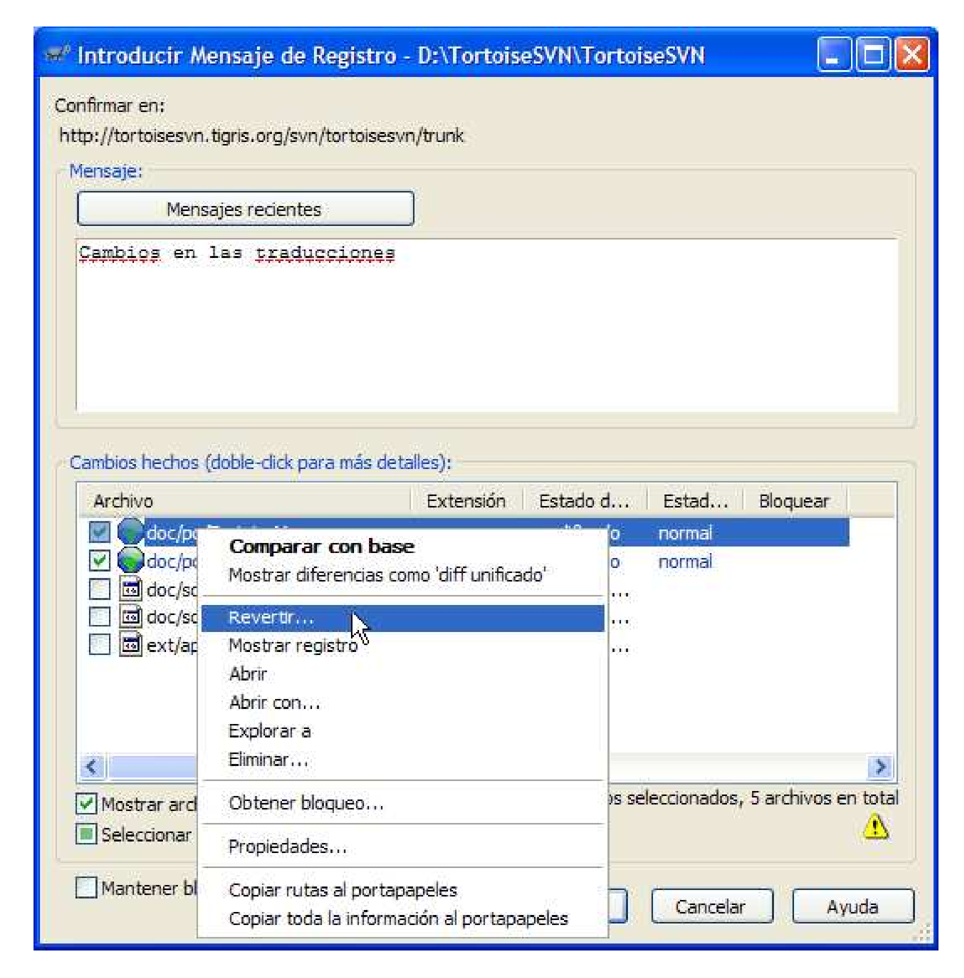
NOTA: Es recomendable obtener únicamente la parte *trunk* del árbol de directorios. Si se especifica la ruta padre del árbol de directorios en la URL, al final se acabará con un disco duro lleno porque se obtendrá una copia del árbol completo del repositorio, incluyendo cada rama y etiqueta del proyecto.

4.3 Enviar cambios al repositorio

Enviar los cambios que se han hecho al repositorio se conoce como confirmar los cambios. Pero antes de confirmar hay que estar seguro de que la copia de trabajo está actualizada. Se puede o bien ejecutar *TortoiseSVN > Actualizar* directamente, o bien ejecutar *TortoiseSVN > Comprobar Modificaciones* primero, para ver qué se ha cambiado localmente o en el servidor. Si la copia de trabajo está actualizada y no hay conflictos, ya se está preparado para confirmar los cambios. Para ello:

1. Seleccionar los ficheros y/o carpetas que desee confirmar.

2. Seleccionar *TortoiseSVN > Confirmar...*



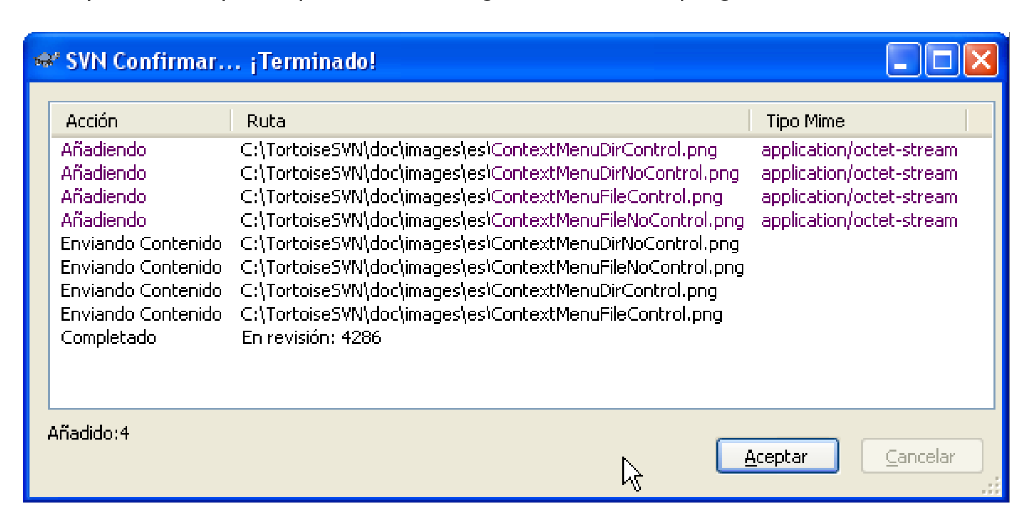
3. El diálogo de confirmación mostrará todos los ficheros cambiados, incluso los ficheros añadidos, borrados o no versionados:

    - Si no se desea que un fichero cambiado se confirme, simplemente se desmarcará ese fichero.
    - Si se desea incluir un fichero no versionado, se marcará para añadirlo a la confirmación.
    - Los ítems que han sido cambiados a una ruta de repositorio diferente también se indican utilizando un marcador.

4. Haciendo doble “clic” en cualquier fichero modificado en el diálogo de confirmaciones, se lanzará la herramienta externa de diferenciar para mostrar los cambios.

5. Habrá que asegurarse de introducir un mensaje de registro que describa los cambios que se están confirmando. Esto ayudará a saber qué ocurrió y cuando según se navegue por los mensajes de registro del proyecto en el futuro. El mensaje puede ser tan extenso o escueto como se desee; muchos proyectos tienen directrices sobre qué debe incluirse en ellos, el idioma que debe utilizarse, y a veces incluso un formato estricto.

6. Tras pulsar “Aceptar” aparece un diálogo mostrando el progreso de la confirmación:



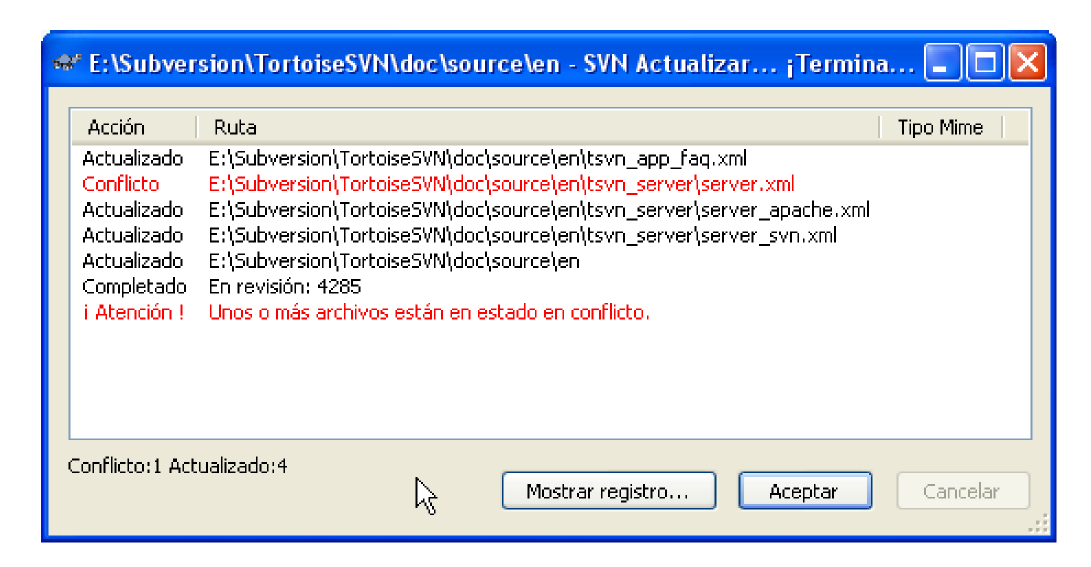
4.4 Actualizar la copia de trabajo con los cambios de otros

Es necesario asegurarse de que los cambios que hacen los demás se incorporen a la copia de trabajo local. El proceso de incorporar los cambios desde el servidor a la copia de trabajo local se conoce como actualización. La actualización puede hacerse en ficheros sueltos, en un conjunto de ficheros, o recursivamente en jerarquías completas de directorios. Para actualizar:

1. Seleccionar los ficheros y/o directorios que se deseen.

2. Hacer “clic” con el botón derecho y seleccionar *TortoiseSVN > Actualizar* en el menú contextual del explorador.

3. Aparecerá una ventana con el progreso de la actualización según se ejecuta. Los cambios que los demás hayan hecho se fusionarán con los ficheros de la copia local, guardando cualquier cambio que se hubiese hecho en los mismos ficheros de la copia local. El repositorio no se ve afectado por una actualización.



1. Si se obtiene algún conflicto durante una actualización (esto puede suceder si los demás han cambiado las mismas líneas del mismo fichero y esos cambios no concuerdan), el diálogo muestra esos conflictos en rojo. Se puede hacer doble "clic" en esas líneas para iniciar la herramienta de fusión externa para resolver los conflictos.

2. Cuando se completa la actualización, el diálogo de progreso le muestra un resumen con el número de ítems actualizados, añadidos, eliminados, en conflicto, etc. bajo la lista de ficheros.

3. El comando *Actualizar* estándar no tiene opciones y simplemente actualiza la copia de trabajo a la revisión HEAD (*) del repositorio, lo que es el caso de uso más común. Si se desea más control sobre el proceso de actualización, se debería utilizar *TortoiseSVN* > *Actualizar* a la revisión.... Esto permite actualizar la copia de trabajo a una revisión específica, no sólo a la más reciente.

NOTA: Actualizar a la revisión puede ser útil a veces para ver cómo estaba el proyecto en un momento anterior en su historia. Pero en general, actualizar ficheros individuales a una revisión anterior no es una buena idea, ya que deja la copia de trabajo en un estado inconsistente.

(*) La revisión HEAD del repositorio es la última revisión de un fichero o una carpeta en el repositorio. En contraposición a ella existe la revisión BASE. La revisión base actual de un fichero o una carpeta en su copia de trabajo. Esta es la revisión en la que estaba el fichero o la carpeta cuando se hizo la última operación de obtener, actualizar o confirmar. La revisión BASE normalmente no equivale a la revisión HEAD.

4.5 Resolver conflictos

De vez en cuando, se obtendrá un conflicto cuando se actualizan los ficheros desde el repositorio. Un conflicto ocurre cuando dos o más desarrolladores han hecho cambios en las mismas líneas de un fichero. Dado que Subversion no sabe nada de su proyecto, delega la resolución de los conflictos en los desarrolladores. Cuando se informa de un confícto, se debería abrir el fichero en cuestión, y buscar líneas que empiecen con el texto <<<<<<<. El área conflictiva se marca así:

```
<<<<<<< nombre-del-fichero
sus cambios
=======
código fusionado del repositorio
```

>>>>>>> revisión

Además, para cada fichero en conflicto Subversion deja tres ficheros adicionales en su directorio:

- **nombre-del-fichero.ext.mine**: Este es el fichero tal y como estaba en la copia de trabajo antes de que se actualizara la copia de trabajo, esto es, sin marcadores de conflicto. Este fichero tiene sus últimos cambios en él y nada más.

- **nombre-del-fichero.ext.rREV-ANTIGUA**: Este es el fichero que era la revisión BASE antes de que se actualizara la copia de trabajo. Esto es, el fichero que se obtuvo antes de empezar a hacer los últimos cambios.

- **nombre-del-fichero.ext.rREV-NUEVA**: Este es el fichero que el cliente de Subversion acaba de recibir desde el servidor del que se actualizó la copia de trabajo. Este fichero corresponde a la revisión HEAD del repositorio.

Entonces:

1. Se puede o bien lanzar una herramienta externa de fusiones / editor de conflictos con el menú contextual *TortoiseSVN > Editar Conflictos* o bien utilizar otro editor manualmente para resolver el conflicto. Se debe decidir cómo tiene que quedar el código, hacer los cambios necesarios, y grabar el fichero.

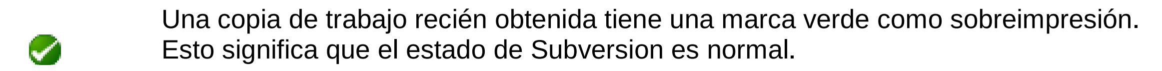
2. Después, se debe ejecutar el comando *TortoiseSVN > Resolver* y confirmar las modificaciones al repositorio. El comando *Resolver* realmente no resuelve el conflicto. Simplemente elimina los ficheros *nombre-del-fichero.ext.mine* y *nombre-del-fichero.ext.r**, dejando confirmar los cambios.

Si se tienen conflictos con ficheros binarios, Subversion no intentará mezclar dichos ficheros por si mismo.

Se puede utilizar el comando *Resuelto para múltiples ficheros* si se pulsa con el botón derecho en la carpeta padre y se selecciona *TortoiseSVN > Resuelto...* Esto mostrará un diálogo con todos los ficheros en conflicto dentro de esa carpeta, y permitirá seleccionar cuáles marcar como resueltos.

4.6 Obtener información de estado

Mientras se está trabajando en la copia de trabajo a menudo será necesario saber qué ficheros se han cambiado/añadido/borrado o renombrado, o incluso qué ficheros han sido cambiados y confirmados por los demás. Para ello son útiles los iconos sobreimpresionados:



Una copia de trabajo recién obtenida tiene una marca verde como sobreimpresión. Esto significa que el estado de Subversion es normal.

En cuanto se empiece a editar un fichero, el estado cambia a modificado y el icono sobreimpresionado cambia entonces a una marca de exclamación roja. De esta forma se puede ver fácilmente qué ficheros se han cambiado desde la última vez que se actualizó la copia de trabajo, y que necesitan ser confirmados.

Si durante una actualización ocurre un conflicto, el icono cambia a un signo de exclamación amarillo.

Si se ha establecido la propiedad *svn:needs-lock* en un fichero, Subversion pone a ese fichero como de sólo-lectura hasta que obtenga un bloqueo en dicho fichero. Los ficheros de sólo-lectura tienen esta sobreimpresión para indicarle que se debe obtener un bloqueo antes de que se pueda editar ese fichero.

Si se ha obtenido un bloqueo sobre un fichero, y el estado de Subversion es normal, este icono sobreimpresionado recordará que se debería liberar el bloqueo si no se está utilizando para permitir a los demás que puedan confirmar sus cambios al fichero.

Este icono muestra que algunos ficheros o carpetas dentro de la carpeta actual se han marcado para ser borrados del control de versiones, o bien que falta un fichero bajo control de versiones dentro de una carpeta.

El signo más indica que el fichero o carpeta está programado para ser añadido al control de versiones.

4.7 Diálogo de registro de revisiones

Para cada cambio que se haga y se confirme, se deberá proporcionar un mensaje de registro de dicho cambio. Así se podrá averiguar después qué cambios se hicieron y por qué, y se tendrá un registro detallado para el proceso de desarrollo.

El diálogo de Registro de revisiones recopila todos esos mensajes de registro y se los enseña. La pantalla se divide en tres paneles:

- El panel superior muestra una lista de revisiones donde se confirmaron cambios a los ficheros/carpetas. Este sumario incluye la fecha y la hora, la persona que confirmó la revisión y el inicio del mensaje de registro. Las líneas azules indican que algo se ha copiado a esta línea de desarrollo (quizás desde una rama).

- El panel medio muestra el mensaje de registro completo para la revisión seleccionada.

- El panel inferior muestra una lista de todos los ficheros y carpetas que se cambiaron como parte de la revisión seleccionada.

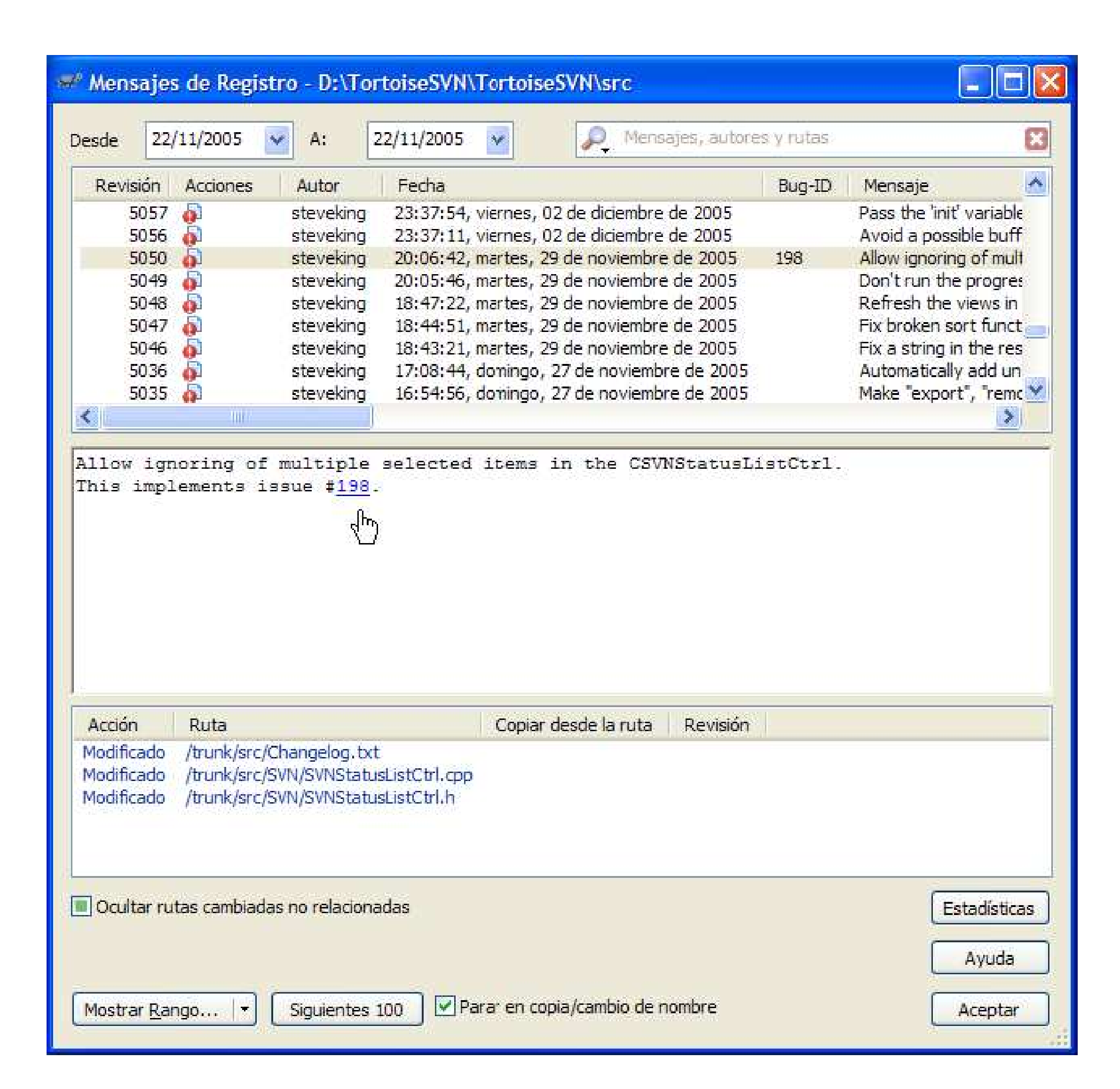
Se puede invocar el diálogo de Registro de revisiones desde:

- El submenú contextual de TortoiseSVN.

- La página de propiedades.

- El diálogo de Progreso después de que se termine una actualización. En ese caso el diálogo de Registro sólo mostrará aquellas revisiones que cambiaron desde la última actualización.

La imagen siguiente muestra el aspecto del diálogo de registro de revisiones:



Como se puede apreciar en la imagen, los comentarios pueden enlazar con bug trackers como Mantis, y de hecho, se pueden definir hooks para cerrar incidencias en Mantis según se actualicen versiones en el repositorio.

4.8 Ver diferencias

Uno de los requisitos más comunes en el desarrollo de proyectos es ver qué ha cambiado. Se puede querer ver las diferencias entre dos revisiones del mismo fichero, o las diferencias entre dos ficheros separados. TortoiseSVN provee una herramienta integrada llamada TortoiseMerge para ver las diferencias entre ficheros de texto.

4.8.1. Diferencias de ficheros

Cambios locales

Si se desea ver qué cambios se han hecho en la copia de trabajo, simplemente se utiliza el menú contextual del explorador y se selecciona *TortoiseSVN > Diferenciar*.

Diferenciar con otra rama/etiqueta

Si se desea ver qué ha cambiado en el tronco (si se está trabajando en una rama) o en una rama específica (si se está trabajando en el tronco), se puede utilizar el menú contextual del explorador. Simplemente manteniendo la tecla Mayúsculas mientras se hace "clic" con el botón derecho en el fichero. Luego, seleccionando *TortoiseSVN > Diferenciar con URL*. En el siguiente diálogo, se especifica la URL del repositorio con la que se quiere comparar el fichero local.

También se puede utilizar el navegador de repositorios y seleccionar dos árboles para diferenciar, quizás dos ramas, o una rama/etiqueta y el tronco. Ahí, el menú contextual permite compararlos utilizando *Comparar revisiones*.

Diferenciar desde una revisión anterior

Si se desea ver las diferencias entre una revisión en concreto y la copia de trabajo, se utiliza el diálogo *Registro de Revisiones*, se seleccionará la revisión de interés, y luego se seleccionará *Comparar con la copia de trabajo* desde el menú contextual.

Diferenciar entre dos revisiones antiguas

Si se desea ver las diferencias entre dos revisiones que ya se confirmaron, se utiliza el diálogo *Registro de Revisiones* y se seleccionarán las dos revisiones que se desea comparar (utilizando el modificador habitual Ctrl). Luego se selecciona *Comparar revisiones* desde el menú contextual.

Si se hizo esto desde el historial de revisiones de una carpeta, aparece un diálogo *Comparar Revisiones*, mostrando una lista de ficheros cambiados en esa carpeta.

Todos los cambios hechos en una confirmación

Si se desea ver los cambios hechos a todos los ficheros en una revisión en particular de una vez, se puede utilizar la salida *diff unificado* (formato de parche GNU). Esto sólo muestra las diferencias con unas pocas líneas de contexto. Es más difícil de leer que una comparación visual de ficheros, pero mostrará todos los cambios juntos. Desde el diálogo *Registro de Revisiones*, se selecciona la revisión de interés, y luego se selecciona *Mostrar Diferencias como Diff Unificado* desde el menú contextual.

Diferencias entre ficheros

Si se desea ver las diferencias entre dos ficheros diferentes, se puede hacer directamente en el explorador seleccionando ambos ficheros (utilizando el modificador habitual Ctrl). Luego desde el menú contextual del explorador se selecciona *TortoiseSVN > Diferenciar*.

Diferencias entre un fichero/carpeta en la copia de trabajo y una URL

Si se desea ver las diferencias entre un fichero en la copia de trabajo, y un fichero en cualquier repositorio de Subversion, se puede hacer directamente en el explorador seleccionando el fichero y pulsando la tecla Mayúsculas mientras se hace "clic" con el botón derecho del ratón para obtener el menú contextual. Se selecciona *TortoiseSVN > Diferenciar con URL*. Se puede hacer lo mismo para una carpeta de copia de trabajo. TortoiseMerge muestra esas diferencias de la misma forma que muestra un fichero de parche - una lista de ficheros cambiados que se pueden ver de uno en uno.

Diferencias con información de autoría

Si se desea ver no sólo las diferencias sino también el autor, revisión y la fecha en la que se hicieron los cambios, se pueden combinar los informes de diferencias y de autoría desde dentro del diálogo del historial de revisiones.

Diferencias entre carpetas

Las herramientas proporcionadas con TortoiseSVN no permiten ver las diferencias entre jerarquías de directorios. Pero si se tiene una herramienta externa que soporte esta funcionalidad, se puede utilizar.

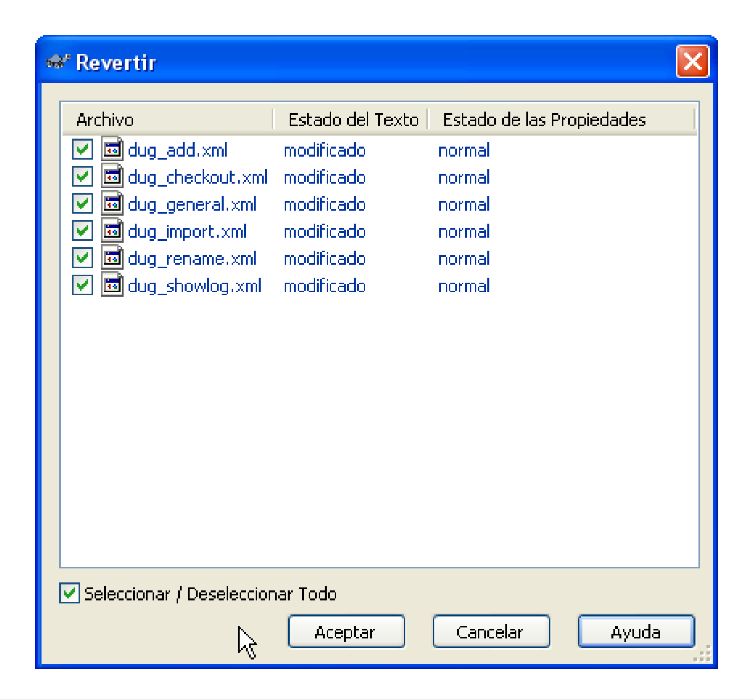
4.9 Añadir nuevos ficheros y directorios

Si se han creado nuevos ficheros y/o directorios durante el proceso de desarrollo, se necesitará añadirlos también al control de código. Para ello se seleccionan los ficheros y/o directorios y se utiliza *TortoiseSVN > Añadir...*

Después de que se añadan los ficheros o directorios al control de código, el fichero aparece con una sobreimpresión de icono añadido que significa que primero se debe confirmar la copia de trabajo para que esos ficheros y directorios estén disponibles para otros desarrolladores.

4.10 Deshacer cambios

Si se desea deshacer todos los cambios que se han hecho en un fichero desde la última actualización, se necesita seleccionar el fichero, hacer "clic" con el botón derecho para sacar el menú contextual, y luego seleccionar el comando *TortoiseSVN > Revertir*. Aparecerá un diálogo que muestra los ficheros que se han cambiado y que se puede revertir. Se seleccionarán los que se desee revertir y se pulsará "Aceptar".



4.11 Limpiar

Si un comando de Subversion no puede completarse de forma correcta, quizás por problemas en el servidor, la copia de trabajo puede quedarse en un estado inconsistente. En ese caso se deberá utilizar *TortoiseSVN > Limpiar* en la carpeta. Es una buena idea hacerlo en la rama superior de la copia de trabajo.

4.12 Crear ramas o etiquetas

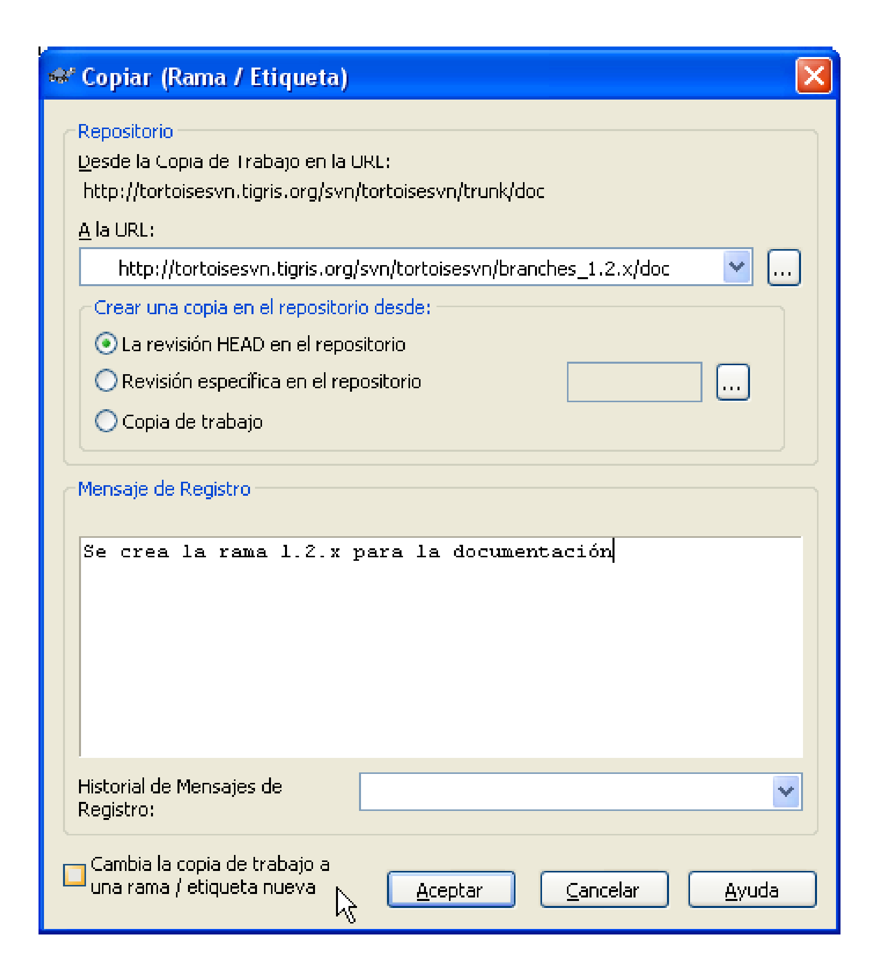
Si se ha importado un proyecto con la estructura de directorios recomendados, crear una rama o una etiqueta es muy simple:

Se selecciona la carpeta en la copia de trabajo de la que se desea hacer una rama o una etiqueta, y luego se selecciona el comando *TortoiseSVN > Rama/Etiqueta...* La URL de destino por defecto para la nueva rama será la URL de origen en la que se basa la copia de trabajo. Se necesitará editar esa URL con la nueva ruta para la rama/etiqueta. Así que en vez de

```
http://svn.collab.net/repos/NombreDelProyecto/trunk
```

se usará algo como

```
http://svn.collab.net/repos/NombreDelProyecto/tags/Version_1.10
```



Ahora se debe elegir el origen de la copia. Existen tres opciones:

- **Revisión HEAD en el repositorio**: La nueva rama se copia directamente en el repositorio desde la revisión HEAD. No se necesita transferir datos desde la copia de trabajo, y la rama se crea muy rápidamente.

- **Revisión específica en el repositorio**: La nueva rama se copia directamente en el repositorio, pero se puede elegir una versión anterior. De nuevo no se transfieren datos desde la copia de trabajo, y la rama se crea muy rápidamente.

- **Copia de trabajo**: La nueva rama es una copia idéntica de la copia de trabajo local. Si se han cambiado algunos ficheros a una revisión anterior en la copia de trabajo, o si se han hecho cambios locales, esto es exactamente lo que irá a la copia. Naturalmente, esta clase de etiquetado complejo conlleva transferir datos desde la copia de trabajo al repositorio si no existe ya allí.

Si se desea que su copia de trabajo se cambie automáticamente a la rama recién creada, se utiliza la casilla *Cambiar la copia de trabajo a la nueva rama/etiqueta*, asegurándose primero de que la copia de trabajo no contenga modificaciones. Si las tiene, esos cambios se mezclarán en la copia de trabajo de la rama cuando se haga el cambio.

Se pulsará "Aceptar" para confirmar la nueva copia al repositorio, proporcionando un mensaje de registro. Hay que tener en cuenta que:

- La copia se crea dentro del repositorio.
- Crear una Etiqueta o una Rama no afecta a la copia de trabajo.

4.13 Fusionar

Mientras que las ramas se utilizan para mantener líneas de desarrollo separadas, en alguna etapa se tendrán que fusionar los cambios hechos en una rama de vuelta en el tronco, o viceversa.

Un punto importante a recordar es que *Fusionar* está muy relacionado con *Diferenciar*. El proceso de fusión trabaja generando una lista de diferencias entre dos puntos del repositorio, y aplicando esas diferencias a la copia de trabajo. Por ejemplo, si se desea fusionar los cambios hechos en la revisión N, entonces se tendrá que comparar la revisión N con la revisión (N-1). Para hacerlo más sencillo, cuando se utilice *Mostrar Registro* para seleccionar un rango de revisiones a fusionar, TortoiseSVN hará el ajuste automáticamente.

En general es una buena idea realizar fusiones en una copia de trabajo sin modificar. Si se han hecho otros cambios en la copia de trabajo, habrá que confirmarlos primero. Si la fusión no funciona como se espera, se puede querer revertir los cambios, y el comando Revertir descartará todos los cambios, incluidos cualquiera que se haya hecho antes de la fusión.